

What is claimed is:

1 1. A system for providing out-of-band notification of service
2 changes, comprising:
3 a cluster framework into a layered architecture, comprising:
4 an application layer comprising at least one of applications and
5 middleware supporting the applications;
6 a database instance resource group interoperating with the
7 application layer and comprising a database instance providing services; and
8 a monitor associated with the database instance resource group and
9 exporting an out-of-band interface to the database instance resource group;
10 a notification mechanism generating an UP service notification from the
11 cluster framework upon service availability and generating a DOWN service
12 notification from the cluster framework upon service non-availability.

1 2. A system according to Claim 1, further comprising:
2 a planned operation interface incorporated into the application layer; and
3 the notification mechanism generating a COMING UP service notification
4 responsive to an instruction received through the planned operation interface and
5 generating a GOING DOWN service notification responsive to a further
6 instruction received through the planned operation interface.

1 3. A system according to Claim 1, further comprising:
2 a global services daemon interfaced to the database instance resource
3 group; and
4 the notification mechanism generating a DOWN service notification for
5 the services on a failed database instance; generating a COMING UP service
6 notification from the global services daemon responsive to a recovering database
7 instance and generating an UP service notification from the global services
8 daemon responsive to a recovered database instance.

1 4. A system according to Claim 1, further comprising:

2 at least one of a remote procedure call interface and an event interface
3 provided to the database instance resource group.

1 5. A system according to Claim 1, further comprising:
2 a resilient set of cluster frameworks comprising an active node and one or
3 more standby nodes.

1 6. A system according to Claim 5, wherein the resilient cluster
2 framework executes a node failover to the active node.

1 7. A system according to Claim 5, wherein the resilient cluster
2 framework executes a node failover to one such standby node.

1 8. A system according to Claim 1, further comprising:
2 a non-resilient set of cluster frameworks comprising an active node.

1 9. A system according to Claim 1, wherein the resilient cluster
2 framework terminates service on a failed node responsive to a DOWN service
3 notification.

1 10. A system according to Claim 1, wherein the resilient cluster
2 framework resumes service on a recovered node responsive to an UP service
3 notification

1 11. A system according to Claim 1, wherein the resilient cluster
2 framework effects a switchover to a standby node responsive to a COMING UP
3 service notification.

1 12. A system according to Claim 1, wherein the application layer pre-
2 connects to a standby node responsive to one of a COMING UP service
3 notification and an UP service notification.

1 13. A method for providing out-of-band notification of service
2 changes, comprising:
3 structuring a cluster framework into a layered architecture, comprising:

4 an application layer comprising at least one of applications and
5 middleware supporting the applications;
6 a database instance resource group interoperating with the
7 application layer and comprising a database instance providing services; and
8 a monitor associated with the database instance resource group and
9 exporting an out-of-band interface to the database instance resource group;
10 generating an UP service notification from the cluster framework upon
11 service availability; and
12 generating a DOWN service notification from the cluster framework upon
13 service non-availability.

1 14. A method according to Claim 13, further comprising:
2 incorporating a planned operation interface into the application layer;
3 generating a COMING UP service notification responsive to an instruction
4 received through the planned operation interface; and
5 generating a GOING DOWN service notification responsive to a further
6 instruction received through the planned operation interface.

1 15. A method according to Claim 13, further comprising:
2 providing a global services daemon interfaced to the database instance
3 resource group;
4 generating a DOWN service notification for the services on a failed
5 database instance;
6 generating a COMING UP service notification from the global services
7 daemon responsive to a recovering database instance; and
8 generating an UP service notification from the global services daemon
9 responsive to a recovered database instance.

1 16. A method according to Claim 13, further comprising:
2 providing at least one of a remote procedure call interface and an event
3 interface to the database instance resource group.

1 17. A method according to Claim 13, further comprising:

2 configuring a resilient set of cluster frameworks comprising an active node
3 and one or more standby nodes.

1 18. A method according to Claim 17, further comprising:
2 executing a node failover to the active node.

1 19. A method according to Claim 17, further comprising:
2 executing a node failover to one such standby node.

1 20. A method according to Claim 13, further comprising:
2 configuring a non-resilient set of cluster frameworks comprising an active
3 node.

1 21. A method according to Claim 13, further comprising:
2 terminating service on a failed node responsive to a DOWN service
3 notification.

1 22. A method according to Claim 13, further comprising:
2 resuming service on a recovered node responsive to an UP service
3 notification

1 23. A method according to Claim 13, further comprising:
2 effecting a switchover to a standby node responsive to a COMING UP
3 service notification.

1 24. A method according to Claim 13, further comprising:
2 pre-connecting to a standby node responsive to one of a COMING UP
3 service notification and an UP service notification.

1 25. A computer-readable storage medium holding code for performing
2 the method according to Claim 13.

1 26. A system for communicating service change events in a cluster
2 framework environment, comprising:

3 a plurality of service change events for communication between a plurality
4 of nodes, comprising:
5 an UP service change event;
6 a DOWN service change event;
7 a COMING UP service change event; and
8 a GOING DOWN service change event;
9 a remote procedure call interface from a database instance in a database
10 stack executing on one such node; and
11 a notification mechanism publishing at least one such service change event
12 from the database instance.

1 27. A system according to Claim 26, further comprising:
2 a further notification mechanism receiving the one such service change
3 event at one or more other nodes.

1 28. A system according to Claim 26, further comprising:
2 a cluster service within the database stack.

1 29. A system according to Claim 26, further comprising:
2 a planned interface within the database stack.

1 30. A system according to Claim 26, further comprising:
2 a global services daemon with listener within the database stack.

1 31. A system according to Claim 26, further comprising:
2 a cluster service processing a multiple instance failover from the one such
3 node to one or more other nodes.

1 32. A system according to Claim 26, further comprising:
2 a cluster service processing a single instance failover to the one such node.

1 33. A system according to Claim 26, further comprising:
2 a cluster service processing a switchover from the one such node to one or
3 more other nodes.

1 34. A method for communicating service change events in a cluster
2 framework environment, comprising:
3 defining a plurality of service change events for communication between a
4 plurality of nodes, comprising:
5 an UP service change event;
6 a DOWN service change event;
7 a COMING UP service change event; and
8 a GOING DOWN service change event;
9 exporting a remote procedure call interface from a database instance in a
10 database stack executing on one such node; and
11 generating a notification publishing at least one such service change event
12 from the database instance.

1 35. A method according to Claim 34, further comprising:
2 receiving the one such service change event at one or more other nodes.

1 36. A method according to Claim 34, further comprising:
2 providing a cluster service within the database stack.

1 37. A method according to Claim 34, further comprising:
2 providing a planned interface within the database stack.

1 38. A method according to Claim 34, further comprising:
2 providing a global services daemon with listener within the database stack.

1 39. A method according to Claim 34, further comprising:
2 processing a multiple instance failover from the one such node to one or
3 more other nodes.

1 40. A method according to Claim 34, further comprising:
2 processing a single instance failover to the one such node.

1 41. A method according to Claim 34, further comprising:

2 processing a switchover from the one such node to one or more other
3 nodes.

1 42. A computer-readable storage medium holding code for performing
2 the method according to Claim 34.

1 43. A method for detecting a failure of a first process, the method
2 comprising the steps of:

3 establishing a first connection between said first process and a second
4 process;

5 monitoring status of said first process to determine whether said first
6 process has failed; and

7 in response to determining that said first process has failed, notifying said
8 second process that said first process has failed;

9 wherein a second connection, that is different from said first connection, is
10 used to notify said second process of said failure of said first process failure.

1 44. A method according to Claim 43, wherein:

2 the step of establishing a first connection between said first process and a
3 second process includes the step of establishing a first connection between an
4 application server and a database instance;

5 the step of monitoring includes the step of monitoring status of said
6 database instance; and

7 the step of notifying said second process that said first process has failed
8 includes the step of causing an out-of-band break to be sent to said application
9 server.

1 45. A method for detecting a failure of a first process, the method
2 comprising the steps of:

3 establishing a first connection between said first process and a second
4 process, wherein said first connection is established using one or more sockets;

5 monitoring status of said first process to determine whether said first
6 process has failed;

7 detecting that said first process has failed, wherein failure of said first
8 process does not cause closure of said one or more sockets; and
9 in response to detecting that said first process has failed, causing an out-
10 of-band notification to be sent to said second process, wherein said out-of-band
11 notification is not sent over said first connection.

1 46. A system for detecting a failure of a first process, the system
2 comprising:
3 a first computer component and a second computer component that are
4 configured to communicate over a first connection; and
5 a fault monitoring component that is configured to monitor the status of a
6 first process associated with said first computer component and in response to
7 detecting a failure of said first process, causing an out-of-band notification to be
8 sent to said second computer component, wherein said out-of-band notification is
9 sent over a second connect that is distinct from said first connection.

20250919 10:26:54